C++ pour la robotique

C++ - Programmation Orientée Objet

Sébastien Rothhut

Programmation Orientée Objet

Objet : structure de données contenant des attributs et des fonctions

Classe : la définition du type d'un objet

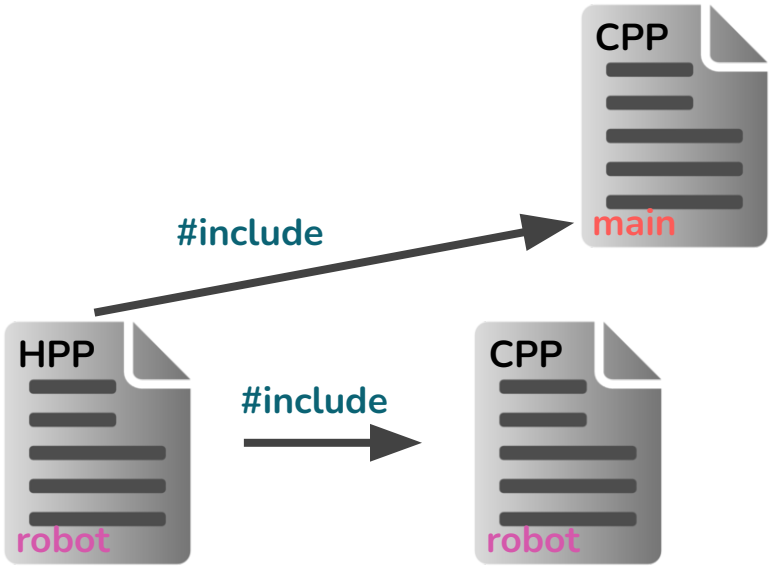
Interface : ensemble des fonctions-membres d'une classe

Programmation Orientée Objet

La programmation orientée objet c’est donc :

- Déclarer des classes (données et fonctions-membres) : interface avec le monde extérieur
- Définir le contenu des fonctions-membres
- Instancier des objets à partir de ces classes
- Effectuer des opérations sur ces objets

Programmation Orientée Objet



1- Déclaration

2- Définition

3- Utilisation

Déclaration d’une classe pour notre robot

Un robot :

- Capteurs
- Actionneurs
- Etat interne
- Qui apporte satisfaction à son utilisateur

Une machine à café !

Déclaration d’une classe

Attributs de la classe **MachineACafe** :

- un réservoir d’eau (**float**)
- un capteur de température (**float**)
- une température cible **(float)**
- une réserve de café (objet café avec variété et quantité disponible)
- un indicateur si un café est en cours (**boolean**)

Déclaration d'une classe

```
class Cafe {
    enum Variete {
        ARABICA, ROBUSTA
    };
    Variete variete;
    int quantite;
};
```

```
class MachineACafe {
    float niveauEau;
    float temperature;
    const float tempCible;
    bool cafeEnCours = false;
    Cafe reserveCafe;
};
```

les Enums

- Liste finie d'éléments constants
- Définit un nouveau type pour lequel on peut déclarer des variables
- Défini de cette façon : `enum Variete {ARABICA, ROBUSTA};`
- Type sous-jacent entier (**int**) par défaut
- Prend le namespace de la classe dans lequel il est déclaré
- Utilisation : `Cafe::Variete v = Cafe::ARABICA;`

les Enums

Exercice : écrire une fonction qui prend un enum du type Variete défini dans la classe Cafe, et qui retourne la chaîne de caractères correspondante

les Enums

```
string afficheVarieteCafe(Cafe::Variete variete) {
     switch (variete) {
     case Cafe::ARABICA:
          return "ARABICA";
     case Cafe::ROBUSTA:
          return "ROBUSTA";
     default:
          return "";
     }
}
```

Déclaration d'une classe

Fonctions-membres :

- `void chaufferMachine()`
- `void faireCoulerCafe(Cafe::Variete variete)`
- `void faireUnCafe(Cafe::Variete variete)`

Déclaration d'une classe

```
class MachineACafe {
    float niveauEau;
    float temperature;
    const float tempCible;
    bool cafeEnCours = false;
    Cafe reserveCafe;

    void chaufferMachine();
    void faireCoulerCafe(Cafe::Variete variete);
    void faireUnCafe(Cafe::Variete variete);
};
```

Public ou privé ?

Déclaration d'une classe

```
class MachineACafe {
    public:
        void faireUnCafe(Cafe::Variete variete);

    private:
        float niveauEau;
        float temperature;
        const float tempCible;
        bool cafeEnCours = false;
        Cafe reserveCafe;
        void chaufferMachine();
        void faireCoulerCafe(Cafe::Variete variete);
};
```

Constructeur

- Public ou privé
- Pas de type de retour
- Nom de la classe
- Peut prendre des paramètres
- L’objet est utilisable

Destructeur

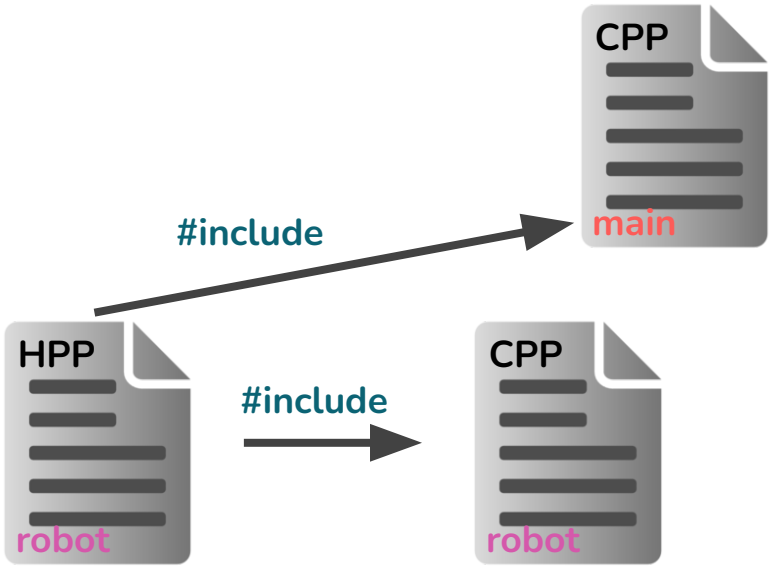
- Public ou privé
- Pas de type de retour
- Nom de la classe préfixé par ~
- Pas de paramètre
- Appelé automatiquement à la fin du cycle de vie de l’objet

Déclaration d'une classe

```
class MachineACafe {
     public:
          MachineACafe();
          void faireUnCafe(Cafe::Variete variete);
          ~MachineACafe();

     private:
          float niveauEau;
          float temperature;
          const float tempCible;
          bool cafeEnCours = false;
          Cafe reserveCafe;
          void chaufferMachine();
          void faireCoulerCafe(Cafe::Variete variete);
};
```

Définition des fonctions membres



1- Déclaration

2- Définition

3- Utilisation

Définition des fonctions membres

```
MachineACafe::MachineACafe() : tempCible(70.0) {}

MachineACafe::MachineACafe(float temp) : tempCible(temp) {}

MachineACafe::~MachineACafe() {}
```

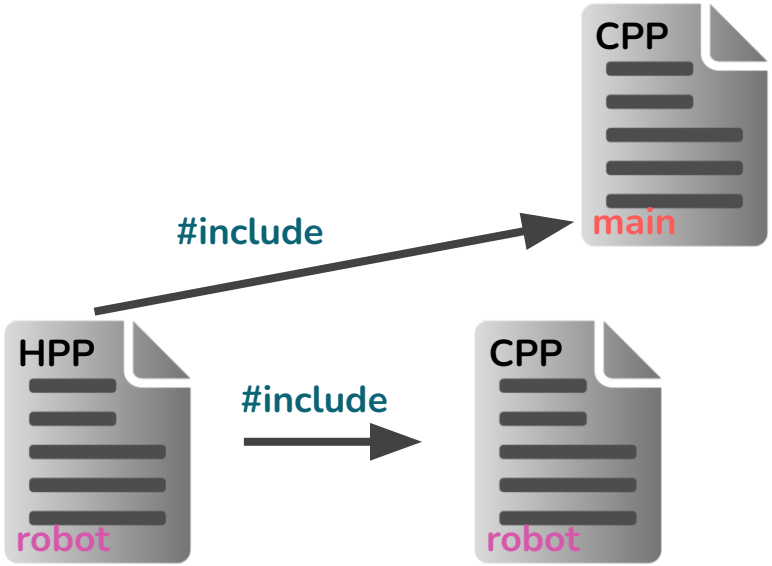
Définition des fonctions membres

```
void MachineACafe::faireUnCafe(Cafe::Variete variete) {
     chaufferMachine();
     faireCoulerCafe(variete);
}

void MachineACafe::chaufferMachine() {
     cout<<"En chauffe..."<<endl;
     cout<<"On atteint "<<tempCible<<" degrés"<<endl;
}

void MachineACafe::faireCoulerCafe(Cafe::Variete variete) {
     cout<<"Je fais couler un "<<afficheVarieteCafe(variete)<<endl;
}
```

Utilisation



1- Déclaration

2- Définition

3- Utilisation

Utilisation

```
int main() {
    MachineACafe maMachine;
    maMachine.faireUnCafe(Cafe::ARABICA);

    return 0;
}
```

Utilisation

```
En chauffe...
On atteint 70 degrés
Je fais couler un ARABICA
```